Description of the Methodology for the Generation of Accuracy Scenarios for Acceptance Testing of the URET CCLD

Robert D. Oaks
Mike M. Paglione
Dr. Hollis F. Ryan
J. Scott Summerill

Working Draft

January 27, 2000

Traffic Flow Management Branch, ACT-250
William J. Hughes Technical Center
Atlantic City International Airport, NJ 08405

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

The Federal Aviation Administration (FAA) is in the process of contracting the Lockheed Martin Corporation Air Traffic Management Division (LMATM) to develop and deploy a Conflict Probe Decision Support Tool. The deployment is limited to seven En Route Air Traffic Control Centers to meet the FAA's Free Flight Phase 1 objective. The limited deployment of the Conflict Probe application is called the User Request Evaluation Tool Core Capability Limited Deployment (URET CCLD). The URET CCLD application is based on the MITRE developed URET Daily Use system installed in Indianapolis and Memphis Centers.

The FAA has tasked the Traffic Flow Management Branch, ACT-250, at the FAA W. J. Hughes Technical Center at Atlantic City to supply LMATM scenarios of realistic air traffic to perform acceptance testing of their system. In particular, these scenarios are to support the accuracy testing and will be used to verify the accuracy requirements of URET CCLD.

AOS-610, in conjunction with ACT-250 and MITRE has collected air traffic data from the Indianapolis (ZID) and Memphis (ZME) Air Route Traffic Control Centers (ARTCCs). This data will be modified to produce the test scenarios. The data will be modified by shifting the start times of aircraft flights and possibly by cloning selected flights. These modifications are made to induce encounters between the aircraft in the test scenarios, but maintain the actual profiles the aircraft originally flew.

1.2 Purpose

This document describes the methodology ACT-250 is using to generate workload scenarios for accuracy testing the User Request Evaluation Tool (URET) Core Capability Limited Deployment (CCLD) software being developed by Lockheed Martin, Air Traffic Management Division. A companion document -- *Description of Acceptance Test Scenarios for the User Request Evaluation Tool (URET) / Core Capability Limited Deployment (CCLD)* [13] -- describes how the scenarios of air traffic generated from air traffic data collected from the ZID and ZME ARTCCs will be analyzed and characterized.

1.3 Scope

The scope of this document is to describe how ACT-250 will use recorded field data to generate the workload scenarios. This includes describing the Oracle database tables that will be used and the software processes that will be used to populate and use these tables to create the scenario files. For this delivery, this document contains numerous TBDs (To Be Determined) and TBSs (To Be Supplied). This is because this is a working draft, which will evolve during the development of this methodology so that at the completion of this project this document will document the methodology thoroughly and accurately.

1.4 Document Organization

This document is organized into six sections. Section 2 identifies the problem, specifying both the input and output requirements and their formats. Section 3 presents ACT-250's solution to the problem. Sections 4 and 5 contain a list of references and a list of tables, respectively.

2 Description of the Problem

ACT-250 was tasked to develop the scenario files to be used by Lockheed Martin for accuracy testing of the URET CCLD software. This accuracy testing is designed to determine (1) whether or not the URET CCLD performs as well as the URET prototype developed by MITRE and (2) whether or not its accuracy as a Decision Support Tool (DST) is adequate to support Air Traffic Controllers. [Reference ???] The testing philosophy is discussed more thoroughly in *Description of Acceptance Test Scenarios for the User Request Evaluation Tool (URET) / Core Capability Limited Deployment (CCLD)*. [13]

The accuracy test will include running two instances of the URET CCLD application in interfacility mode using the Memphis (ZME) and Indianapolis (ZID) ARTCC adaptations for the May 20, 1999 chart cycle. The accuracy test will be performed only on the ZME instance. The ZID application is only needed to provide the interfacility functionality.

The scenario files used for the accuracy test will be based on recorded field data, but modified in order to satisfy a number of constraints. The scenario files delivered to Lockheed Martin will be ASCII files containing messages based on the Common Message Set (CMS).

ACT-250 will run versions of these same scenarios, based on the HCS 3.20 Patch (P320), through the URET Daily Use system developed by MITRE. The data collected during these runs will be measured by ACT-250 and provided to AUA-200, who will implement these results into the test specification. [Reference ??].

ACT-250 is also required to characterize the information contained in the RUC weather files. This is discussed in in *Description of Acceptance Test Scenarios for the User Request Evaluation Tool (URET) / Core Capability Limited Deployment (CCLD)*. [13]

2.1 Field Data Input

The input data for this effort consists of field data recorded at the Memphis (ZME) and Indianapolis (ZID) ARTCCs on May 26th and 27th, 1999. This data includes:

- URET Recorder Data - ACT-250 will use the files recorded by the ZME and ZID URET Daily Use system as the source of HCS messages

- Other Field Data - The field data recorded for this effort will be using the HCS 3.20 Patch (P320). Since the scenarios for the URET CCLD will be based on the Common Message Set (CMS), the URET recorded data will not be sufficient. System Analysis Recording (SAR) tapes created by the ZME and ZID HCSs and other, yet to be determined, data sources will be used to fulfill this requirement.

- Weather Data - The hourly Rapid Update Cycle (RUC) weather files will be needed for both the URET and the URET CCLD scenario playback runs. This data will not need to be manipulated by ACT-250. National Weather Service (NWS) RUC 236 files will be delivered to Lockheed Martin. WARP Stage 0 RUC 211 files will be used to characterize the weather data and as input to URET playback runs.

2.2 Scenario Outputs

ACT-250 will create the scenario files in two formats: Common Message Set (CMS) and P320 formatted ASCII files.

2.2.1 CMS Formatted Scenario Files

The scenario files provided to Lockheed-Martin by ACT-250 are based on the CMS messages defined in the *ARTCC Host Computer System/ Air Traffic Management Applications Interface Requirements Document*[8]. Lockheed Martin has identified the following subset of those messages as those required in the URET CCLD scenario files.

- Flight Plan Information Message (FH)
- Flight Plan Amendment Information Message (AH)
- Cancellation Information Message (CL)
- Departure Information Message (DH)
- Sector Assignment Status Information Message (SH)
- Aircraft Identification Amendment Information Message (IH)
- Hold Information Message (HH)
- Drop Track Information Message (RH)
- Interim Altitude Information Message (LH)
- Progress Report Information Message (PH)
- Expected Departure Time Information Message (ET)
- Track Information Message (TH)

The scenario file is a binary file in which each CMS message is prefaced with two headers:

- **GMT_Header.** The GMT_Header consists of a single, 22-byte field containing Greenwich Mean Time encoded in ASCII as "*GMT yyyymmdd.hhmmssmmm*." I.e., the three ASCII characters "GMT," followed by a space, followed by the four-digit year, two-digit month, a period, the two-digit hour, two-digit minute, two-digit second, and three digit millisecond. For example "*GMT 19990526.134555904*" represents May 26, 1999 at 13:45:55.904.

- **HID_Header**. The HID_Header consists of a single 26-byte field containing the *message_length*, its *sequence_number*, an unused field, and a *time_stamp*. The *message_length* is a four-byte, signed integer representing the number of bytes in the message, not including the headers. The *sequence_number* is a four-byte, signed integer. The unused field consists of four bytes. The time_stamp is a 14-byte field containing an ASCII time stamp encoded in ASCII as "*yyyymmddhhmmss*," which represents the year, month, day, hour, minute, and second as defined for the GMT_Header.

- **Message.** The message contains the EBCDIC encoded fields as specified in the IRD in which each message

- begins with a Source ID (4 bytes formatted in EBCDIC) and a LRC (2 binary bytes),
- contains message dependent, space separated fields,
- and ends with a LRC (2 binary bytes) and an End of Message (1 byte containing the bit sequence 10110001).

The following subsections describe the contents and source of data for each of the CMS formatted messages. Table 2-1 summarizes which of the CMS messages will be included in the CMS formatted scenario file and the criteria for including the specific message.

Table 2-1: CMS Message Generation

NAS Field Ref	Message Description	When generated in CMS scenario file
FH	Flight Plan Information	A CMS FH message will be generated for the first recorded P320 FP message for a flight that had track data.
AH	Flight Plan Amendment	A CMS AH message will be generated for each recorded P320 FP message following the first recorded P320 FP message for a flight that had track data.
CL	Cancellation Information	A CMS CL message will be generated for each recorded P320 RS message for a flight that had track data.
DH	Departure Information	A CMS DH message will be generated when it is determined that a flight is for a departure aircraft.
SH	Sector Assignment Information	A CMS SH message will be generated for each recorded P320 OS message.
IH	Aircraft Identification Amendment	A CMS IH message will be generated whenever it is detected that flight's CID has changed.
HH	Hold Information	A CMS HH message will be generated for each recorded P320 HM message for a flight that had track data.
RH	Drop Track Information	A CMS RH mess age will be generated for each recorded P320 QX message for a flight that had track data.
LH	Interim Altitude	A CMS LH message will be generated for each recorded P320 QQ message for a flight that had track data.
PH	Progress Report Information	A CMS LH message will be generated for each recorded P320 PR message for a flight that had track data..
ET	Expected Departure Time Information	TBD[1]
TH	Track Information	A CMS TH message will be generated for each recorded P320 TT message.

[1] There is no comparable P320 message. If required for accuracy testing a CMS ET message will be constructed algorithmically or manually.

2.2.1.1 CMS FH Message

A Flight Plan Information Message (FH) message will be generated in the CMS formatted scenario file at a time relative to when an FP message was provided in the recorded field data. Its contents and the source of data is presented in Table 2-2.

Table 2-2: CMS Flight Plan Information Message

Contents:	Source of Data
Flight Identification	*flight_table.acid* + "/" + *flight_table.cid*
Aircraft Data	"1/" + *flight_table.ac_type* + *flight_table.ac_equip*
Beacon Code	*flight_table.b_code*
Speed	*flight_plan.speed*
Coordination Fix	*flight_plan.coord_fix*
Coordination Time	*flight_plan.coord_time*
Assigned or Requested Altitude	*flight_plan.altitude*
Route	*flight_plan.route*
(Uncombined FPA Containing the First Postable Fix)	*flight_plan.first_post_fix*
(Uncombined FPA Containing the First PAR Fix)	*flight_plan.first_PAR_fix*
(Adapted Route Indicator)	*flight_plan.adapted_route*
(PDR/PDAR Alphanumerics and Successive Field 10 Element)	*flight_plan.PDR_PDAR*
(PAR Alphanumerics and Preceding Field 10 Element)	*flight_plan.PAR*
(Remarks)	*flight_plan.remarks*

2.2.1.2 CMS AH Message

A Flight Plan Amendment Message (AH) message will be generated in the CMS formatted scenario file at a time relative to when an AM messages was provided in the recorded field data. The contents of the AH message is identical with the Flight Plan Information Message (see Table 2-2).

2.2.1.3 CMS CL Message

A Cancellation Information Message (CL) message will be generated in the CMS formatted scenario file after the completion of an aircraft track. Its contents and the source of data is presented in Table 2-3.

Table 2-3: CMS Cancellation Information Message

Contents:	Source of Data
Flight Identification	*flight_table.acid* + "/" + *flight_table.cid*
Departure Point	Extracted from *flight_plan.route*
Destination	Extracted from *flight_plan.route*

2.2.1.4 CMS DH Message

The Departure Information Message (DH) will be generated in the CMS formatted scenario file for each departure track. Its contents and the source of data are presented in Table 2-4.

Table 2-4: CMS Departure Information Message

Contents:	Source of Data
Flight Identification	*flight_table.acid* + "/" + *flight_table.cid*
Aircraft Data	"1/" + *flight_table.ac_type* + *flight_table.ac_equip*
Departure Point	Extracted from *flight_plan.route*
Actual Departure Time	Provided by software based on *flight.start_time*.
Destination	Extracted from *flight_plan.route*
(ETA)	This optional field will not be provided.

2.2.1.5 CMS SH Message

TBD

Table 2-5: CMS Sector Assignment Information Message

Contents:	Source of Data
Airspace Assignment	TBD

2.2.1.6 CMS IH Message

TBD

Table 2-6: CMS Aircraft Identification Amendment Message

Contents:	Source of Data
Flight Identification (old)	TBD
Flight Identification (new)	TBD
Departure Point	TBD
Destination	TBD

2.2.1.7 CMS HH Message

The Hold Information Message (HH) will be generated in the CMS formatted scenario file as the HM message was provided in the recorded field data.

Table 2-7: CMS Hold Information Message

Contents:	Source of Data
Flight Identification	TBD
Hold Data	TBD

2.2.1.8 CMS RH Message

The Drop Track Information Message will be generated in the CMS formatted scenario file as the QX message was provided in the recorded field data.

Table 2-8: CMS Drop Track Information Message

Contents:	Source of Data
Flight Identification	TBD

2.2.1.9 CMS LH Message

The Interim Altitude Information Message will be generated in the CMS formatted scenario file as the QQ message was provided in the recorded field data.

Table 2-9: CMS Interim Altitude Message

Contents:	Source of Data
Interim Altitude	TBD
Flight Identification	TBD

2.2.1.10 CMS PH Message

TBD

Table 2-10: CMS Progress Report Information Message

Contents:	Source of Data
Flight Identification	TBD
Position	TBD
Progress Report Data	TBD

2.2.1.11 CMS ET Message

TBD

Table 2-11: CMS Expected Departure Time Information Message

Contents:	Source of Data
Flight Identification	TBD
Estimated Departure Clearance Time	TBD

2.2.1.12 CMS TH Message

The Track Information Message (TH) will be generated in the CMS formatted scenario file for each track point.

Table 2-12: CMS Track Information Message

Contents:	Source of Data
Flight Identification	*flight_table.acid* + "/" + *flight_table.cid*
Ground Speed	*track_table.ground_speed*
Assigned Altitude	TBD
Reported Altitude	*track_table.altitude*
Controlling Facility/Sector	TBD
Receiving Facility/Sector	TBD
Track Data	*track_table.x_pos* and *track_table.y_pos* converted to lat/long.
Track Velocity	TBD
Coast Indicator	TBD

2.2.2 P320 Formatted ASCII Files

ACT-250 will also need to provide the same scenarios in a format suitable for input to the URET test bed in the TFM Lab. This is necessary in order to recalibrate the URET CCLD Specification[9]. The P320 formatted scenario messages will be generated into an ASCII file in a format compatible with a number of MITRE tools. This format is documented in the *XEVAL Users Manual* and in Appendix A of *Algorithmic Evaluation Process and Tools for the User Request Evaluation Tool*.[5][6]

3 Solution

ACT-250 is developing an Air Traffic database representing a model of the aircraft flights captured in the field. The data recorded by the URET Daily Use system is being used to initially create and populate the tables within this database. Additional data, required to meet the additional needs of the CMS format, will be extracted from other sources and inserted into the database tables independently.

ACT-250 will develop a scenario generation program, which accesses the tables within the Air Traffic Database, to create two ASCII scenario files; one using the CMS format and the other using the P320 format.

ACT-250 will use the P320 formatted ASCII file as input to existing processes and procedures which will populate the Conflict Probe database. (ACT-250 will need to modify some of these processes and procedures.) The data in this database will then be evaluated using software developed by ACT-250 to determine information relevant to the constraint criteria that must be satisfied by the scenarios. This evaluation will include information about trajectory accuracy, conflict prediction accuracy, and scenario characteristics.

ACT-250 will develop a process by which the start times of the aircraft contained in the Air Traffic database can be changed when making the scenario files. This process may be done manually through the update of a table; or it may be done algorithmically (for example through the used of a Genetic Algorithm, see Section 3.2.11.2).

Figure 3-1 presents a data flow diagram showing the files, databases, processes, documents, and data flows ACT-250 will use in this process. The processes (shown as circles) and databases are described in this section as follows:

- Air Traffic Database; see Section 3.1.1
- Conflict Probe Database; see Section 3.1.2
- Extract Message Data; see Section 3.2.1
- Extract Other Data; see Section 3.2.2
- Characterize Weather; see Section 3.2.3
- Make URET CCLD Scenarios; see Section 3.2.4
- Run URET in TFM Lab; see Section 3.2.5
- Parse URET Data; see Section 3.2.6
- Scenario Processing; see Section 3.2.7
- Evaluate Conflict Probe Accuracy; see Section 3.2.8
- Evaluate Trajectory Accuracy; see Section 3.2.9
- Characterize Air Traffic; see Section 3.2.10
- Modify Field Data; see Section 3.2.11

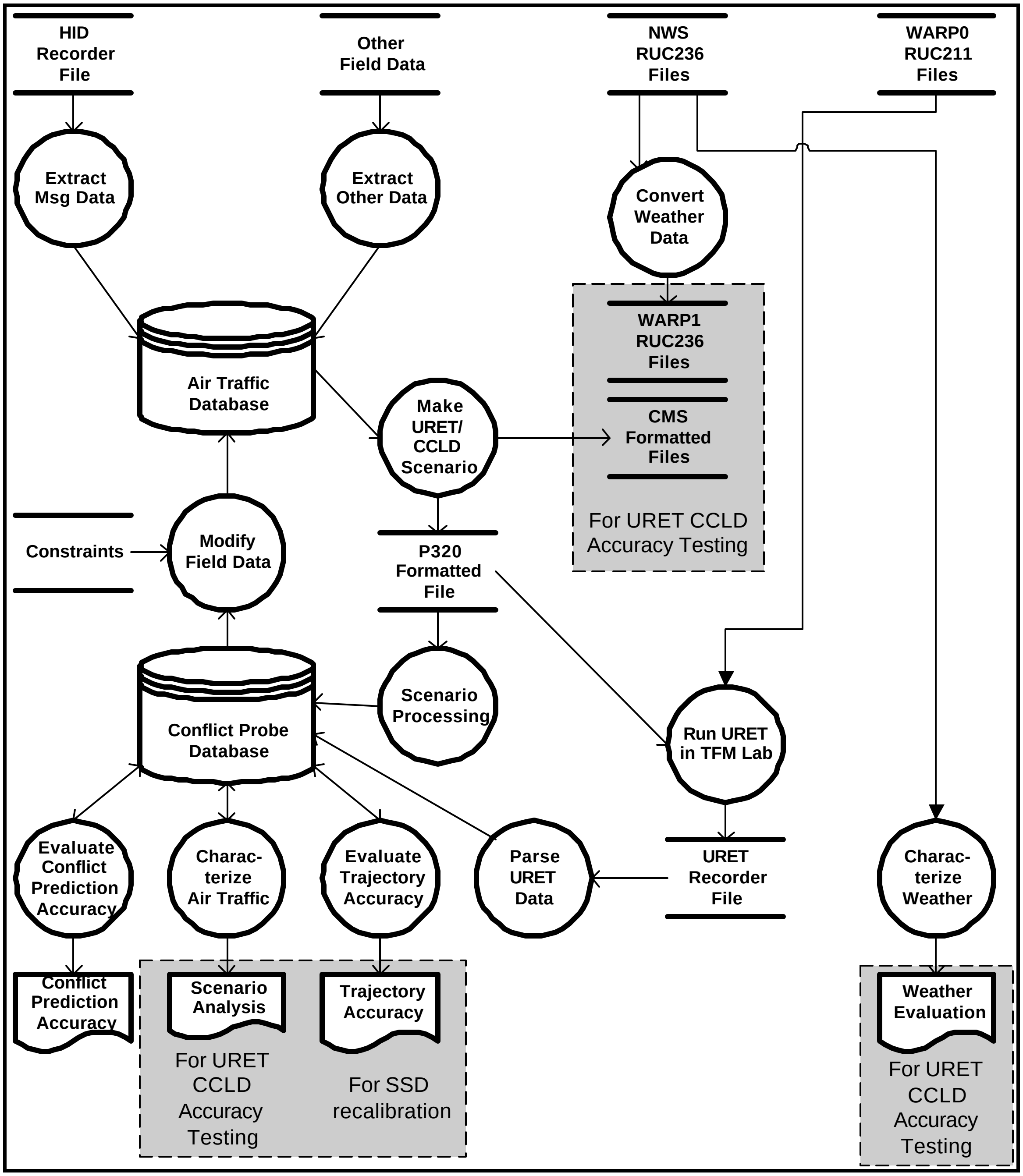

Figure 3-1: Data Flow for CCLD Accuracy Scenario Generation

3.1 Databases

3.1.1 Air Traffic Database

The Air Traffic Database utilizes the TFM Laboratory's Oracle V8.04 database. The following subsections describe the Air Traffic Database tables. The fields contained within these tables are described in detail in the P320 and CMS Interface Requirements Documents. [7][8]

3.1.1.1 fd_airspace Table

The *fd_airspace* table contains the constants required for x-y to lat-long coordinate conversion for a specific ATC facility. These data items are inserted into *fd_airspace* when it is created by a UNIX script.

Table 3-1: fd_airspace Table

Col Name	Data Type	Units	Description	Source
facility_id (key)	varchar2(4)	n/a	FAA facility designator (e.g., ZID, ZME)	Established by software.
lat_tang_deg	number(3)	deg	Latitude at point of tangency	Established by software.
lat_tang_min	number(2)	min	Latitude at point of tangency	Established by software.
lat_tang_sec	number(2)	sec	Latitude at point of tangency	Established by software.
lon_tang_deg	number(3)	deg	Longitude at point of tangency	Established by software.
lon_tang_min	number(2)	min	Longitude at point of tangency	Established by software.
lon_tang_sec	number(2)	sec	Longitude at point of tangency	Established by software.
x_tang	number(10,6)	nm	X coordinate at point of tangency	Established by software.
y_tang	number(10,6)	nm	Y coordinate at point of tangency	Established by software.
e_radius	number(11,6)	nm	Conformal earth radius	Established by software.

3.1.1.2 fd_rtix Table

The *fd_rtix* table contains the pref-route information contained in the URET RTIX table. These data items are inserted into *fd_rtix* when it is created by a program written specifically to extract this information from the RTIX table generated by URET.

Table 3-2: fd-rtix Table

Col Name	Data Type	Units	Description	Source
facility_id (key)	varchar2(4)	n/a	FAA facility designator (e.g., ZID, ZME)	Established by software.
adapt_rte_index (key)	number(4)	n/a	Adapted route index	Extracted from the RTIX table by software.
pref_rte_name (key)	varchar2(7)	n/a	Pref-route name.	Extracted from the RTIX table by software.
pref_rte_type	varchar2(5)	n/a	Pref-route type.	Extracted from the RTIX table by software.
index	number(4)	n/a		Extracted from the RTIX table by software.

3.1.1.3 fd_data_id Table

The *fd_data_id* table contains information identifying the data sets recorded at the ATC facility. Except for the *populated* flag, the data items in *fd_data_id* are inserted by the user. The *populated* flag is set by the message extraction software once the data from a HID file has been inserted.

Table 3-3: fd_data_id Table

Col Name	Data Type	Units	Description	Source
data_set (key)	number(3)	n/a	Data set identifier	User provided.
facility_id	varchar2(4)	n/a	FAA facility designator (e.g., ZID, ZME)	User provided.
testdate	varchar2(9)	date	Date field data was recorded	User provided.
basetime	number(9,3)	sec	The *start_time* fields in the *fd_flight* table are relative to this base time.	User provided.
populated	number(1)	n/a	Flag indicating whether the *fd_flight*, *fd_track* tables for this data have been populated by the message extraction software. (0=no, 1=yes)	Software assigned.
remarks	varchar2(40)	n/a	Descriptive remarks	User provided.

3.1.1.4 fd_run Table

The *fd_run* table contains information identifying the data sets to be used for a specific run. These data items are inserted into *fd_run* by the *ext* process (see Section 3.2.1) and the *Odo* process (see Section 3.2.11).

Table 3-4: fd_run Table

Col Name	Data Type	Units	Description	Source
run_no (key)	number(3)	n/a	Run number	Software assigned.
data_set_p	number(3)	n/a	Primary data set identifier	User provided.
data_set_s	number(3)	n/a	Secondary data set identifier	User provided.
remarks	varchar2(40)	n/a	Descriptive remarks	Software assigned.

3.1.1.5 fd_sector_asgn Table

The *fd_sector_asgn* table contains information associated with the sector assignment message. These data items are inserted into *fd_sector_asgn* by the *ext* process (see Section 3.2.1).

Table 3-5: fd_sector_asgn Table

Col Name	Data Type	Units	Description	Source
data_set	number(3)	n/a	Data set number	Software assigned.
relative_time	number(8,3)	sec	Time relative to the basetime	Extracted from the OS message.
sector_no	varchar2(2)	n/a	Sector number	Extracted from the OS message.
fpa_no	varchar2(4)	n/a	Fixed posting area	Extracted from the OS message.
fpa_sub_no	varchar2(4)	n/a	Subjugate_fixed posting area	Extracted from the OS message.

3.1.1.6 fd_flight Table

The *fd_flight* table contains the static information related to a flight. It also contains the mapping of ACID/CID to *ac_no*, which is the primary key in many of the other tables.

Table 3-6: fd_flight Table

Col Name	Data Type	Units	Description	Source
data_set (key)	number(3)	n/a	Data set number	Software assigned.
acid (key)	varchar2(7)	n/a	Flight identifier	Extracted from the first message extracted for this aircraft.
cid (key)	varchar2(3)	n/a	Computer id	Extracted from the first message extracted for this aircraft.
ac_no	number(4)	n/a	Key used to access other tables	Software assigned.
start_time	number(8,3)	sec	Track start time	The *start_time* is defined as the time of the first track point for the flight.
b_code	varchar2(4)	n/a	Beacon code	Extracted from the first FP message for this aircraft.
heavy_ind	varchar2(2)	n/a	Heavy indicator	Extracted from the first FP message for this aircraft.
ac_type	varchar2(4)	n/a	Type of aircraft	Extracted from the first FP message for this aircraft.
ac_equip	varchar2(1)	n/a	Airborne equipment qualifier	Extracted from the first FP message for this aircraft.
drop_track_time	number(8,3)	sec	Time, relative to track start time, for drop track message	
rem_strip_time	number(8,3)	sec	Time, relative to track start time, for remove strip message	
run_no (key)	number(3)	n/a	Run number (0 indicates original extracted data)	Software assigned.
delta_time	number(7,3)	sec	Delta time by which to shift the original start time of an aircraft	User provided or software assigned.
cull_flag	number(2)	n/a	Flag indicating whether or not to include flight in scenario generation.	User provided or software assigned.

3.1.1.7 fd_flight_plan Table

The *fd_flight_plan* table contains a history of flight plan and amendment messages for a specific aircraft.

Table 3-7: fd_flight_plan Table

Col Name	Data Type	Units	Description	Source
data_set (key)	number(3)	n/a	Data set number	Software assigned
ac_no (key)	number(4)	n/a	Access key	Software assigned.
fp_no (key)	number(2)	n/a	Sequential number for flight plans to accommodate amendments	Software assigned.
relative_time	number(9,3)	sec	Time relative to track start time	*relative_time* is defined as the difference between the time associated with the entry and the associated track's *start_time*.
speed	varchar2(4)	knots	True air speed	Extracted from the FP message.
coord_fix	varchar2(12)	n/a	Coordination fix	Extracted from the FP message.
coord_type	varchar2(2)	n/a	D, E, or P	Extracted from the FP message.
coord_time	varchar2(4)	hhmm	Coordination time	Extracted from the FP message.
altitude	varchar2(20)	100's ft	Assigned or requested altitude	Extracted from the FP message.
route	varchar2(576)	n/a	Route including destination	Extracted from the FP message.
first_post_fix	varchar2(4)	n/a	Uncombined FPA containing the first postable fix	TBD
first_PAR_fix	varchar2(4)	n/a	Uncombined FPA containing the first PAR fix	TBD
adapted_route	varchar2(12)	n/a	Departure route name ID and/or arrival route name ID	TBD
PDR_PDAR	varchar2(54)	n/a	PDR/PDAR element alphanumerics	TBD
PAR	varchar2(54)	n/a	PAR element alphanumerics	TBD
remarks	varchar2(40)	n/a	Remarks	Extracted from the FP message.

3.1.1.8 fd_track Table

The *fd_track* table contains the individual track points for a flight.

Table 3-8: fd_track Table

Col Name	Data Type	Units	Description	Source
data_set (key)	number(3)	n/a	Data set number	Software assigned
ac_no (key)	number(4)	n/a	Access key	Software assigned.
relative_time	number(9,3)	sec	Time relative to track start time	*relative_time* is defined as the difference between the time associated with the entry and the associated track's *start_time*.
lat_deg	number(3)	deg	Latitude	Extracted from the TT message.
lat_min	number(2)	min	Latitude	Extracted from the TT message.
lat_sec	number(2)	sec	Latitude	Extracted from the TT message.
lon_deg	number(3)	deg	Longitude	Extracted from the TT message.
lon_min	number(2)	min	Longitude	Extracted from the TT message.
lon_sec	number(2)	sec	Longitude	Extracted from the TT message.
altitude	number(6)	feet	Altitude	Extracted from the TT message.
ctl_sector	varchar2(5)	n/a	Controlling sector	Extracted from the TT message.
rcv_sector	varchar2(5)	n/a	Receiving sector	Extracted from the TT message.
nas_velx	number(6,2)	kts	NAS x velocity	Extracted from the TT message.
nas_vely	number(6,2)	kts	NAS y velocity	Extracted from the TT message.

3.1.1.9 fd_int_alt Table

The *fd_int_alt* table contains a history of Interim Altitude messages for a specific aircraft.

Table 3-9: fd_int_alt Table

Col Name	Data Type	Units	Description	Source
data_set (key)	number(3)	n/a	Data set number	Software assigned
ac_no (key)	number(4)	n/a	Access key	Software assigned.
int_alt_no (key)	number(2)	n/a	Sequential number for Interim Altitude messages.	Software assigned.
relative_time	number(9,3)	sec	Time relative to track start time	*relative_time* is defined as the difference between the time associated with the entry and the associated track's *start_time*.
interim_altitude	varchar2(4)	n/a	Interim altitude	Extracted from the QQ message.

3.1.1.10 fd_hold Table

The *fd_hold* table contains a history of Hold messages for a specific aircraft.

Table 3-10: fd_hold Table

Col Name	Data Type	Units	Description	Source
data_set (key)	number(3)	n/a	Data set number	Software assigned
ac_no (key)	number(4)	n/a	Access key	Software assigned.
hold_no (key)	number(2)	n/a	Sequential number for Hold messages.	Software assigned.
relative_time	number(9,3)	sec	Time relative to track start time	*relative_time* is defined as the difference between the time associated with the entry and the associated track's *start_time*.
fix	varchar2(12)	n/a	Fix	Extracted from the HM message.
hold_time	varchar2(4)	n/a	Hold time	Extracted from the HM message.
action	varchar2(2)	n/a	Action	Extracted from the HM message.

3.1.2 Conflict Probe Database

The Conflict Probe Database is an existing Oracle database developed for previous studies done by ACT-250. [1][12] These are described in the *Conflict Probe Data Reduction and Tools Interface Control Document*.[14]

3.2 Software Processes

The software processes used for this task are run in ACT-250's Traffic Flow Management (TFM) Laboratory. This laboratory is composed of a number of networked Sun Ultra SPARC I - 167 MHz workstations using the Solaris Version 2.6 operating system. For data storage there is a 96 GB Digital RAID 450 Storage Array.

The software ACT-250 is developing for this effort consists of UNIX scripts and a number of C/C++ programs. ACT-250 uses gcc, the GNU C/C++ Version 2.7.2.3 compiler, and libg++, the GNU C/C++ Version 2.7.2 libraries. Programs accessing the Oracle database use the Oracle Pro*C/C++ Precompiler Release 8.0.

The processes, shown as circles in Figure 3-1, are summarized in the and described in this subsection.

Table 3-11: Software Processes

Process	Program(s)	Section
Extract Message Data	*ext*	3.2.1.1
Extract Other Data	*getRTIX*	3.2.2.1
Characterize Weather		
Make URET CCLD Scenario Files	*sgp*	3.2.4.1
Run URET in TFM Laboratory		
Parse URET Data		
Scenario Processing		
Evaluate Conflict Probe Accuracy		
Evaluate Trajectory Accuracy		
Characterize Air Traffic		
Modify Field Data	*Odo*	3.2.11.1

3.2.1 Extract Message Data

ACT-250 is developing a program called *ext* that populates the Air Traffic Database from the extracted HCS message data.

3.2.1.1 ext

ext is a C++ program which uses the Oracle Pro*C/C++ Precompiler to create and update the Air Traffic Database. This program uses MITRE's *pc320fileclass* contained within the *320parser* program.[2] This C++ class contains methods that read and isolate the PC320 messages from the HID recorder file.

3.2.1.1.1 *ext* Inputs

ext requests the user to input the name of the HID recorder file, the three character facility ID, and the date that the recording was made. The identified HID recorder file is then processed.

3.2.1.1.2 *ext* Outputs

ext's primary output is the extacted data in the Air Traffic tables. In addition, *ext* records log messages in the *FD_PROC.log* file. The specific tables are:

- *ext* inserts an entry in the *fd_data_id* table for the facility ID and the date.
- *ext* extracts the data from the HID recorder file and populates the *fd_flight*, *fd_run*, *fd_track*, *fd_int_alt*, *fd_flight_plan*, and *fd_hold* tables setting the *data_set* variable to the appropriate newly inserted value.
- As *fd_flight* is populated, *ext* sets the *run_no* and *delta_time* variables to zero. Prior to program termination *ext* inserts a duplicate set of entries with the *run_no* variable set to the next available value. *ext* then inserts an entry in the *fd_run* table setting the *run_no* to the appropriate newly inserted value. This becomes the nominal run for generating a scenario. The entries with *run_no* equal to zero are used by the programs that modify the field data as the entries in which *delta_time* can be modified.

3.2.2 Extract Other Data

ACT-250 will develop special purpose programs or scripts to obtain data and insert it into the Air Traffic Database. To do this, ACT-250 may use existing FAA tools (such as DART) and MITRE tools to obtain this data. These tools include: *getRTIX*:

3.2.2.1 getRTIX

getRTIX is a C++ that … TBS.

3.2.3 Characterize Weather

AOS-610 will develop software that will read the WARP Stage 0 RUC211 Weather data and provide summary information regarding this data.

3.2.4 Make URET CCLD Scenario Files

ACT-250 is developing a program called *sgp* that uses the data contained within the Air Traffic Database to generate scenarios in either the Common Message Set (CMS) or HCS 3.20 Patch (P320) format.

3.2.4.1 sgp

sgp is a C++ program that uses the Oracle Pro*C/C++ Precompiler to access the Air Traffic Database. This process writes ASCII scenario files based on the Air Traffic database model. The

ASCII scenario files will contain time tagged messages records, limited to those messages identified and described in Section 2.2.

3.2.4.1.1 *sgp* Inputs

When launched *sgp* solicits the following information from the user:

- the run number of the run in the *fd_run* table from which to make the scenario.
- if a P320 formatted scenario is to be generated.
- if a CMS formatted scenario is to be generated.

After creating an unsorted scenario file, the script used to launch *sgp* again asks a number of questions in order to post-process the data (e.g., to determine if sorting is required and addition information that may be needed).

3.2.4.1.2 *sgp* Outputs

ext's primary output is a P320 formatted and/or CMS formatted scenario file. In addition, *sgp* records log messages in the *FD_PROC.log* file.

3.2.5 Run URET in TFM Laboratory

ACT-250 will use the P320 Formatted Scenario to run the URET prototype in the TFM Laboratory. This run will verify the characteristics of the scenario.

Many of the processes that ACT-250 will use during this verification were developed for other activities and are documented elsewhere[1][4][10].

3.2.6 Parse URET Data

TBS

3.2.7 Scenario Processing

TBS

3.2.8 Evaluate Conflict Probe Accuracy

TBS

3.2.9 Evaluate Trajectory Accuracy

TBS

3.2.10 Characterize Air Traffic

ACT-250 will develop Oracle PL/SQL scripts and programs that use the Oracle Pro*C/C++ Precompiler to access the Air Traffic Database and summarize key measures that describe the air traffic. This would include measures (such as arrival rates, departure rates, within rates, overflight rates), percentage of aircraft at specific altitude strata, and statistics related to CPA (Closest Point of Approach). This characterization is described in detail in *Description of Acceptance Test Scenarios for the User Request Evaluation Tool (URET) / Core Capability Limited Deployment (CCLD).* [13]

3.2.11 Modify Field Data

The contents of the Air Traffic Database will be extracted from recorded field data; however the scenarios must contain conflicts and other criteria that do not occur in the field data. In order to achieve meet these constraints ACT-250 will modify this data by only changing the start times of the recorded aircraft flights and adjust the occurrence of other flight related data (for example: flight plans and interim altitude messages) relative to this change.

ACT-250 may also insert additional flights (called clones) based on recorded flight paths. This will result in realistic flights, but with the unrealistic aircraft-to-aircraft encounters that are necessary to test the accuracy of URET CCLD.

The techniques used to change change flight start times may include compressing the times of specific field recorded events with respect to some basetime and randomly shifting the times of these events. The process of cloning would require modifying a flight's ACID and CID and determining an appropriate insertion time. Effectively determining how these techniques should be applied will probably require an algorithm. This section discusses the specific software and techniques ACT-250 may use; these include Odo, a program that uses time compression and time shifting, the genetic algorithm, and cloning.

3.2.11.1 Odo

Odo is a C++ program that uses the Oracle Pro*C/C++ Precompiler to access the Air Traffic Database to modify the data items in the *fd_flight* table in order to alter the start times of the flights. *Odo* uses time compression and time shifting to accomplish this alteration.

- **Time Compression** means to apply a constant compression multiplier (C_m) times the difference between the start times of the aircraft tracks and some base time. Figure 3-2 presents this technique. T_b represents the base time, T_0 represents the track start time, and T_i, $i = 1,2,3,\ldots$ represents the track points. With time compression the start time of the original track, T_0, is changed to T'_0 so that $T'_0 = T_b + C_m(T_0 - T_b)$. Since the track data is effected by the winds, the times of the hourly RUC weather updates would be compressed by the same amount. Note with time compression the sequence of aircraft in the scenario is left unchanged.

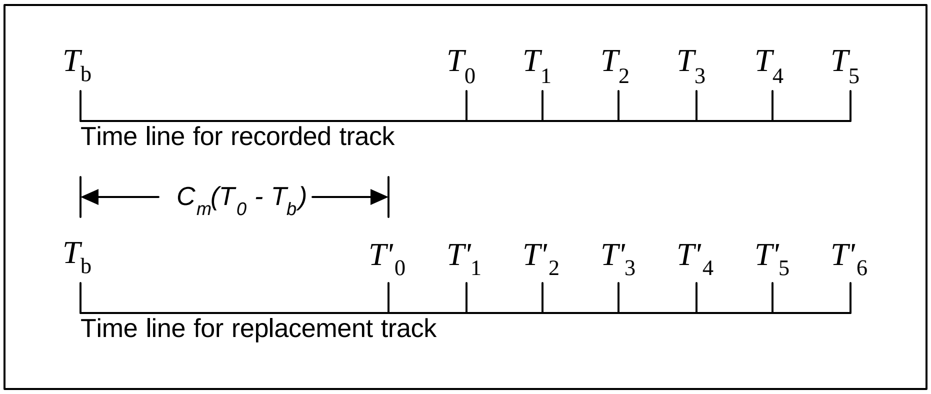

Figure 3-2: Time Compression

- **Time shifting** means to modify the start time of the aircraft tracks by adding a random delta time. Figure 3-3 presents this technique. The start time of the original track, T_0, is changed to T'_0 so that $T'_0 = T_0 - \boldsymbol{r}$, where $\boldsymbol{r}$ is randomly selected from some known frequency distribution (Normal or Uniform). With time shifting there could be a problem associated with the corresponding weather data. Therefore a limit to the amount of time shifting might need to be applied. Note that with time shifting the sequence of aircraft can be changed.

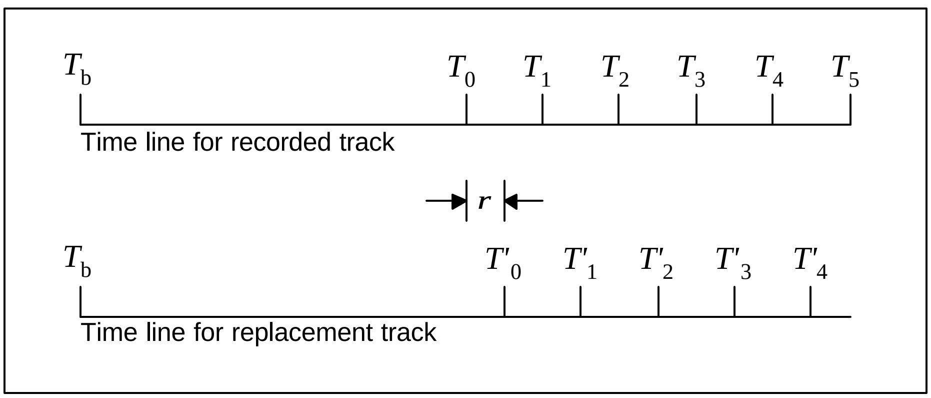

Figure 3-3: Time Shifting

3.2.11.1.1 Odo inputs

Odo requests the following information from the user:
- The *data_set* of the primary site (i.e., the test facility).
- The *data_set* of the secondary site (i.e., the facility which in being used for interfacility testing). This is optional.
- The percentage of compression where 10 would mean a compression of 10% (which would represent a compression multiplier (C_m) of 0.9). An entry of 0 would indicate no time compression.
- The distribution to be used for time shifting where 0 indicates no random time shifting, 1 indicates a normal distribution is to be used, and 2 indicates a uniform distribution is to be used.

- The standard deviation of the normal distribution (in seconds) if normal random time shifting was selected.
- The range of the uniform distribution (low value, high value) if uniform random time shifting was selected.

3.2.11.1.2 Odo outputs

Odo inserts entries:

- in the *fd_run* table setting the *run_no* to the next available value.
- in the *fd_flight* table setting the *run_no* and *delta_time* variables.

3.2.11.2 Genetic Algorithm

A technique ACT-250 has investigated and may implement as a process for algorithmically determining start times for aircraft flights is the Genetic Algorithm (GA). This is a class of algorithms that derive their behavior from a metaphor of the processes of evolution. GAs are appropriate when the solution space is large, not well understood, and when a sufficiently good solution is adequate (i.e.; a global optimum is not required). [3][11]

3.2.11.3 Cloning

Cloning means to insert additional aircraft into a scenario based on existing aircraft. A simple approach to cloning would be to simply insert a duplicate of an existing aircraft track at another point in time. Figure 3-4 presents this technique. In this example, the start time of the cloned flight is set at an arbitrary time denoted T'_0. This technique creates a completely new sequence of tracks and care would need to be taken in order to ensure that the cloned track is inserted at a time that weather would not be a factor.

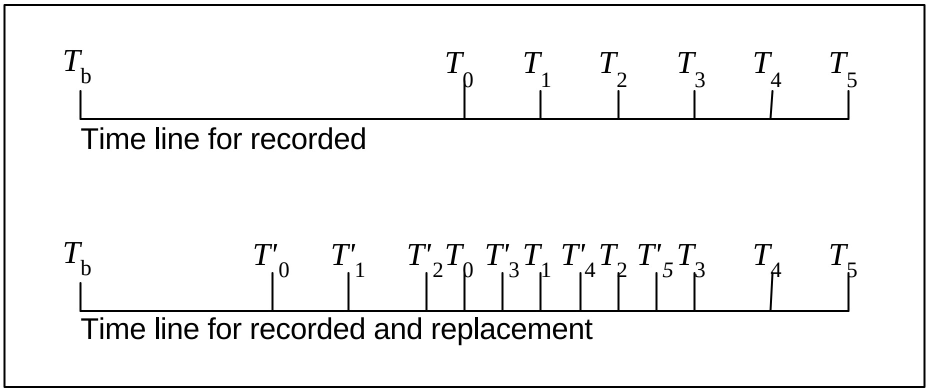

Figure 3-4: Cloning

Cloning would be more effective if the cloned aircraft were inserted in a manner that would benefit the scenario with regards to its constraints. This would require descriptive statistics such as flight type, altitude, direction, and phase of flight has been gathered on existing flights in the database and that duplicate flights be injected based on need to meet encounter requirements.

4 References

[1] Mary Lee Cale, Michael Paglione, Dr. Hollis Ryan, Dominic Timoteo, Robert Oaks, *User Request Evaluation Tool (URET) Conflict Prediction Accuracy Report*, DOT/FAA/CT-TN98/8, April 1998.

[2] Andrew M. DeLiso, *User's Manual for the URET P320 Host Patch Parser*, MITRE, WN 98W0000126, November, 1998.

[3] David E. Goldberg, *Genetic Algorithms in Searching, Optimization, and Machine Learning*, Addison-Wesley, 1989.

[4] Betty Kelly, Mike Paglione, *Conflict Probe Performance (CPP) Analysis Tools - User's Manual Version 1.0a*, August 31, 1998.

[5] Kenneth S. Lindsay, *Xeval Users Manual*, MITRE Technical Report, MTR 97W0000030, June, 1997.

[6] Kenneth S. Lindsay, *Algorithmic Evaluation Process and Tools for the User Request Evaluation Tool*, MITRE Technical Report, MTR 98W0000029, June, 1998.

[7] Federal Aviation Administration, *Interface Requirements Document ARTCC Host Computer System (HCS) / Center TRACON Automation System (CTAS) - User Request Evaluation Tool (URET)*, NAS-IR-82178219, October 5, 1998.

[8] Federal Aviation Administration, *Interface Requirements Document ARTCC Host Computer System (HCS) / Air Traffic Management (ATM) Applications*, NAS-IR-82170001, January 8, 1999.

[9] Lockheed Martin Air Traffic Management, User Request Evaluation Tool (URET) Core Capability Limited Deployment (CCLD) System Specification (SSS), Volume I, Part 2: Conflict Probe (CP), August 21, 1998.

[10] Lockheed Martin Air Traffic Management, SW Tools Need for GFP Scenario Assessment, SIG Issue ICP166, June 21, 1999.

[11] Melanie Mitchell, *An Introduction to Genetic Algorithms*, The MIT Press, Cambridge, MA, 1998.

[12] Mike M. Paglione, Dr. Hollis F. Ryan, Robert D. Oaks, J. Scott Summerill, Mary Lee Cale, *Trajectory Prediction Accuracy Report: User Request Evaluation Tool (URET) / Center-TRACON Automation System (CTAS)*, DOT/FAA/CT-TN99/10, May 1999.

[13] Mike M. Paglion, Dr. Hollis F. Ryan, Scott Summerill, Robert D. Oaks, *Description of Accuracy Scenarios for the Acceptance Testing for the User Request Evaluation Tool (URET) / Core Capability Limited Deployment (CCLD)*, (Review Draft) September 1999.

[14] WJHTC/ACT-250, *Conflict Probe Data Reduction and Tools Interface Control Document*, in progress.

5 Acronyms

ACID	Aircraft Identification
AH	CMS Flight Plan Amendment Message
ARTCC	Air Route Traffic Control Center
ASCII	American Standard Code for Information Exchange
CCLD	Core Capability - Limited Deployment
CID	Computer Identification
CL	CMS Cancellation Information Message
CMS	Common Message Set
CPA	Closest Point of Approach
DART	Data Analysis and Reduction Tool
DH	CMS Departure Information Message
DST	Decision Support Tool
ET	CMS Expected Departure Time Information Message
FAA	Federal Aviation Administration
FH	CMS Flight Plan Information Message
FP	P320 Flight Plan Message
FPA	Fix Posting Area
GB	Gigabyte
GNU	GNU's Not Unix
HCS	Host Computer System
HH	CMS Hold Information Message
HID	Host Interface Device
HR	CMS Route Status Information Message
IH	CMS Aircraft Identification Amendment Message
IRD	Interface Requirements Document
LH	CMS Interim Altitude Message
LMATM	Lockheed Martin Air Traffic Management
MHz	Mega Hertz
NWS	National Weather System
P310	HCS 3.10 patch
P320	HCS 3.20 patch
PAR	Preferential Arrival Route
PDAR	Preferential Arrival and Departure Route
PDR	Preferential Departure Route
PH	CMS Progress Report Information Message
QQ	P320 Interim Altitude Message
RAID	Redundant Array of Inexpensive Disks
RH	CMS Drop Track Information Message
RTIX	TBS
RUC	Rapid Update Cycle
SAR	System Analysis Recording
SH	CMS Sector Assignment Information Message

TBD	To Be Determined
TBS	To Be Supplied
TFM	Traffic Flow Management
TH	CMS Track Information Message
TT	P320 ATM Track/full data block Message
URET	User Request Evaluation Tool
WARP	Weather And Radar Processor
ZID	Indianapolis ARTCC
ZME	Memphis ARTCC